Computer Science 101—Winter 2017
Project Design Document

Elevator Design Document

Friday, February 3, 2017

Version 1.4

David Casperson
Team: *Doppio con Panne*

Contents

Chapter 1

Introduction

This document was created for CPSC 101 in Winter 2017 by Team *Doppio con Panne* with David Casperson as lead author. It was revised in 2020 to add class designs for a few more of the significant classes.

This design document is incomplete. Classes like `Person` *should have the same attention to detail as classes like* `Elevator`*.*

This document gives the preliiminary design of an elevator simulator, whose specification is given in Appendix A.

1.1 Work Distribution to Date

Person	Percentage
David Casperson	66
Dr David Casperson	33
Casper the ghost	1
TOTAL	100

Implementation Plans ... are pretty vague.

1.2 Document History

VERSON		DATE
1.0	initial version	January 30, 2017
1.1	imaginary version	
1.2	imaginary version	
1.3	for CPSC 101 2019	January 23, 2019
1.4	for CPSC 101 2020	December 27, 2019

.

Chapter 2

Design Elements

This chapter contains a list of nouns and list of facts. The next chapter contains detailed design on a per-noun basis.

2.1   List of Nouns

These have been alphabetized and converted to singular form.

- action
- activity
- arrival
- (two-floor office) building
- button
- capacity
- clock
- clock tick
- company
- day
- direction
- elevator
  - bell
  - button
  - door
  - arrival light
- energy
- floor
  - button
    - door
- message
- passenger
- person
- scheduler
- the screen
- (software) simulator
- system
- time

2.2 List of Facts

This list is not sorted.

- the building has two floors and an elevator.

- the simulator simulates the building for a day.

- the simulator has a clock.

- the clock ticks once per second.

- the clock starts at zero.

- the simulator has a scheduler.

- the scheduler starts the day by picking two random times

- the first random time is the time the person on floor 1 arrives.

- the second random time is the time the person on floor 2 arrives.

- the random times are in the range $[5, 20] \cap \mathbb{Z}$.

- When a person arrives she steps onto the floor and pushes the floor button.

- A pressed floor button illuminates.

- The floor button light turns off when the button is reset.

- the elevator has a capacity of one person.

- each floor has a capacity of one person.

- Exiting passengers don't affect capacity.

- When the elevator arrives at a floor, it resets the elevator button.

- When the elevator arrives at a floor, it sounds the elevator bell.

- the elevator bell is inside the elevator.

- After the elevator sounds its bell it signals its arrival to the floor.

- When signaled, the floor resets the floor button.

- When signaled, the floor turns on the floor's elevator arrival light.

- after the floor arrival light turns on the elevator opens the doors.

- after the doors open the elevator's passenger (if any) exits.

- after the doors open the floor's person (if any) enters the elevator.

- after a person enters the elevator, she presses the elevator

button.

- *missing facts about what happens when the elevator door closes*

- presing the elevator button illuminates it.

- the elevator button resets and turns off its light when the elevator arrives on the floor.

- all the activities that happen, from when the elevator reaches a floor until the elevator closes its door, take zero time.

- The elevator takes five seconds to move from one floor to the other.

- Once per second, the simulator provides the time to the scheduler and to the elevator.

- The simulator displays messages on the screen that describe the activities that occur in the system.

Chapter 3

Per Noun Entries

Each entry lists relevant facts for the noun, and if the noun merits it, a brief paragraph description. Assuming that the noun is a candidate for a class, a list of attributes, behaviours, and collaborations follows.

3.1 Action

3.1.1 Facts

- The elevator and the scheduler use time to determine what actions to take.

3.1.2 Summary

Action is too vague to become a class.

3.1.3 Other Notes

. . .

3.2 Activity

Activity appears to be a synonym for **Action**.

3.2.1 Summary

Activity is too vague to become a class.

3.2.2 Other Notes

. . .

3.3 Arrival

3.3.1 Facts

- the first random time is the time the person on floor 1 arrives.
- the second random time is the time the person on floor 2 arrives.
- Exiting passengers don't affect capacity.
- When the elevator arrives at a floor, it resets the elevator button.
- When the elevator arrives at a floor, it sounds the elevator bell.
- After the elevator sounds its bell it signals its arrival to the floor.
- When signaled, the floor resets the floor button.
- When signaled, the floor turns on the floor's elevator arrival light.
- after the floor arrival light turns on the elevator opens the doors.

Arrival   There are two kinds of arrival. Each is associated with a floor. One is when a person appears on a floor. The other is when the elevator arrives at a floor. In the former case, the person does things. In the latter case, the elevator and floor do things in a certain sequence.

3.3.2 Summary

Arrival is not likely a class on its own but might give rise to behaviours in **Person**, **Elevator**, and **Floor**.

3.3.3 Other Notes

. . .

3.4 Bell

3.4.1 Facts

- When the elevator arrives at a floor, it sounds the elevator bell.
- the elevator bell is inside the elevator.

Bell The elevator has a bell, which rings when the elevator arrives at floor.

3.4.2 Attributes

none.

3.4.3 Behaviours

- **ring**.

3.4.4 Collaborations

none.

3.5 Building

3.5.1 Facts

- the building has two floors and an elevator.

Building The building is where the simulation takes place, so presumably a simulation has a building. In turn a building has two floors, and an elevator.

3.5.2 Attributes

- floor 1[1]
- floor 2
- elevator

3.5.3 Behaviours

none

3.5.4 Collaborations

none.

3.5.5 Other Notes

. . .

---

[1]I should probably figure out soon what the getter functions are here. If it's `public Floor getFloor(int n)`, does n range over 0 and 1, or 1 and 2?

Perhaps it should be `public Floor [] getFloors()`.

3.6   Button

3.6.1   Facts

- When a person arrives she steps onto the floor and pushes the floor button.
- A pressed floor button illuminates.
- The floor button light turns off when the button is reset.
- When the elevator arrives at a floor, it resets the elevator button.
- after a person enters the elevator, she presses the elevator button.
- presing the elevator button illuminates it.
- the elevator button resets and turns off its light when the elevator arrives on the floor.

Button   There are two kinds of buttons: floor buttons and elevator buttons. There are three buttons, one on each floor, and one in the elevator. All buttons contain a light which is turned on when the button is pressed, and turned off when the button is reset. Only persons push buttons; only containing objects (floors and elevators) reset buttons.

3.6.2   Attributes

- **light**.

3.6.3   Behaviours

- **push**
- **reset**

3.6.4   Collaborations

none with outside objects. Pushing a button causes it to turn on its light. Resetting a button causes it to turn off its light.

3.6.5   Other Notes

It might be useful to have an `isPushed()` attribute. . . .

3.7 Capacity

3.7.1 Facts

- the elevator has a capacity of one person.
- each floor has a capacity of one person.
- Exiting passengers don't affect capacity.

Capacity   Both floors and the elevator have a capacity of one person. At present this is fixed by design.

3.7.2 Summary

Capacity is not likely a class on its own but might give rise to attributes in **Floor** and **Elevator**.

3.7.3 Other Notes

. . .

3.8 Clock

3.8.1 Facts

- the simulator has a clock.
- the clock ticks once per second.
- the clock starts at zero.
- all the activities that happen, from when the elevator reaches a floor until the elevator closes its door, take zero time.
- Once per second, the simulator provides the time to the scheduler and to the elevator.

Clock   The simulator has a clock, which keeps track of time. The clock is very rudimentary, and keeps track of time in terms of an integer number of seconds, which starts at zero.

3.8.2 Attributes

- time (in integer seconds).

3.8.3 Behaviours

- **reset**.
- **tick**.

3.8.4 Collaborations

none

3.8.5 Other Notes

. . .

3.9 Clock tick

see **Clock**.

3.10 Company

not needed.

3.11 Day

3.11.1 Facts

- the simulator simulates the building for a day.

Day   The simulator has a clock, and simulates the building and elevator for a day.

3.11.2 Summary

Day is not likely a class on its own, but we need to figure out how long a day is (midnight to midnight?).

3.11.3 Other Notes

. . .

3.12 Door

3.12.1 Facts

- after the floor arrival light turns on the elevator opens the doors.
- after the doors open the elevator's passenger (if any) exits.
- after the doors open the floor's person (if any) enters the elevator.
- *missing facts about what happens when the elevator door closes*

Door   There are three doors, one on each floor, and one on the elevator. The elevator door opens only when the elevator is stopped at a floor, and there is a button pushed that necessitates opening the door. The floor door opens only when the elevator is stopped at that floor, and the elevator door opens.

3.12.2 Attributes

- openness.

3.12.3 Behaviours

- open.
- close.

3.12.4 Collaborations

- *none*.

3.12.5 Other Notes

- The elevator door and floor door act simultaneously.

3.13 Direction

3.13.1 Facts

- There are no explicit facts about `Direction`.

Direction   The elevator car can have one of three directions states: `UP` (the elevator is heading upward), `DOWN` (the elevator is heading downward), and `STATIONARY` (the elevator car is at reat at a floor).

3.13.2 Other Notes

- Likely modelled as an `enum`.

3.14 Elevator

3.14.1 Facts

- is part of the the building.
- the elevator has a capacity of one person.
- Exiting passengers don't affect capacity.
- When the elevator arrives at a floor, it resets the elevator button.
- When the elevator arrives at a floor, it sounds the elevator bell.
- the elevator bell is inside the elevator.
- After the elevator sounds its bell it signals its arrival to the floor.
- When signaled, the floor turns on the floor's elevator arrival light.
- after the floor arrival light turns on the elevator opens the doors.
- after the doors open the elevator's passenger (if any) exits.
- after the doors open the floor's person (if any) enters the elevator.
- after a person enters the elevator, she presses the elevator button.
- presing the elevator button illuminates it.
- the elevator button resets and turns off its light when the elevator arrives on the floor.
- all the activities that happen, from when the elevator reaches a floor until the elevator closes its door, take zero time.
- The elevator takes five seconds to move from one floor to the other.
- Once per second, the simulator provides the time to the scheduler and to the elevator.

Elevator   The elevator is part of a building with two floors. The elevator responds to button presses, both for buttons on floors and the button in the elevator. The elevator takes 5 s to move between floors. When the elelvator arrives at a floor, it rings a bell, opens its door, releases its passenger (if any), receives waiting passengers (if any).

3.14.2 Attributes

- location
- direction of travel
- arrival time
- passenger

3.14.3 Behaviours

- open door
- close door
- reset elevator button.
- move.
- receive passenger
- discharge passenger

3.14.4 Collaborations

- The transfer of a passenger from/to a floor involves the floor, the elevator and possibly the passenger.
- The elevator, scheduler, and clock somehow interact so that the elevator takes 5s to travel between floors.

3.14.5 Other Notes

- The elevator *has a* door, button, light, bell, and *is in a* building. Although these are not explicitly mentioned as attributes, there ought to be ways to get at these objects from the elevator.

3.15 Energy

3.15.1 Facts

- we would like to minimize the energy usage.

Energy   The elevator uses energy.

3.15.2 Summary

Energy is not likely to be a class.

3.15.3 Other Notes

. . .

3.16 Floor

3.16.1 Facts

- the building has two floors and an elevator.
- When a person arrives she steps onto the floor and pushes the floor button.
- A pressed floor button illuminates.
- The floor button light turns off when the button is reset.
- each floor has a capacity of one person.
- Exiting passengers don't affect capacity.
- After the elevator sounds its bell it signals its arrival to the floor.
- When signaled, the floor resets the floor button.
- When signaled, the floor turns on the floor's elevator arrival light.
- after the floor arrival light turns on the elevator opens the doors.

Floor   The building contains two floors. Each floor has a capacity to contain one person. Each floor has a floor button, and an elevator arrival light. The scheduler places persons on each floor. Initially Persons leave a floor by getting on the elevator. When a person arrives at her destination floor she vanishes.

3.16.2 Attributes

- surrounding building
- surrounding simulator
- elevator call button
- elevator arrival light
- (floor) door.[2]
- capacity (fixed at one?).
- persons on the floor.

3.16.3 Behaviours

- **acceptPerson**.
- **removePerson**.
- **acceptArrivalSignal**
- **noticeDeparture**
- **resetButton**

3.16.4 Collaborations

- the **Scheduler** and the **Floor** need to co-operate to move a **Person** to the floor.
- the **Elevator** and the **Floor** need to co-operate to move a **Person** from the floor to the elevator.

---

[2]may not be needed?

3.16.5   Other Notes

- the floor can find the elevator by asking the building where it is.

...

3.17 Message

not written.

3.18 Passenger

Passenger appears to be a synonym for **Person** when a person is in an elevator.

3.18.1 Summary

Passenger is more likely an attribute name (for an elevator) than a class.

3.18.2 Other Notes

. . .

3.19 Person

not written.

3.20 Scheduler

not written.

3.21 Screen

not written.

3.22 Simulator

not written.

3.23 System

not written.

3.24 Time

not written.

Appendix A

Simulator Specification

A company intends to build a two-floor office building and equip it with an elevator. The company wants you to develop an object-oriented *software simulator* in Java that models the operation of the elevator to determine whether the elevator suits the company's needs.

Your simulator should include a clock that begins with its time, in seconds, set to zero. The clock ticks (increments the time by one) every second, but it does not keep track of the hours and minutes. Your simulator also should include a scheduler that begins the day by scheduling two times randomly: the time when a person first steps onto floor 1 and the button on that floor to summon the elevator, and the time when a person first steps onto floor 2 and presses the button on that floor to summon the elevator. Each of these times is a random integer in the range from 5 to 20 seconds, inclusive (i.e., 5, 6,7, . . . , 20). When the clock time equals the earlier of these two times, the scheduler creates a person, who then walks onto the appropriate floor and presses the floor button. [*Note*: It is possible that these two randomly scheduled times will be identical, in which case people step onto both floors and press both floor buttons at the same time.] The floor button illuminates, indicating that it has been pressed. [*Note*: The illumination of the floor button occurs automatically when the button is pressed and needs no programming; the light built into the button turns off automatically when the button is reset.] At the beginning of the simulation, the elevator starts the day waiting with its door closed on floor 1. To conserve energy, the elevators moves only when necessary. The elevator alternates directions between moving up and moving down.

For simplicity, the elevator and each of the floors have a capacity of one

person. The scheduler first verifies that a floor is unoccupied before creating a person to walk onto that floor. If the floor is occupied, the scheduler delays creating the person by one second (thus allowing the elevator an opportunity to pick up the person and move to the floor). After a person walks onto a floor, the scheduler creates the next random (between 5 and 20 seconds into the future) for a person to walk onto that floor and press the floor button.

When the elevator arrives at a floor, it resets the elevator button and sounds the elevator bell (which is inside the elevator). The elevator then signals its arrival to the floor. The floor, in response, resets the floor button and turns on the floor's elevator-arrival light. The elevator then opens its door. [*Note*: The door on the floor opens automatically with the elevator door and needs no programming.] The elevator's passenger, if there is one, exits the elevator, and a person, if there is one waiting on that floor, enters the elevator. Although each floor floor has a capacity of one person, assume there is enough room on each floor for a person to wait on that floor while the elevator's passenger exits.

A person entering the elevator presses the elevator button, which illuminates (automatically, without programming) when pressed and turns off when the elevator arrives on the floor and resets the elevator button. [*Note*: Because the building has only two floors, only one elevator button is necessary; this button notifies the elevator to move to the other floor.] Next, the elevator closes its door and begins moving to the other floor. When the elevator arrives at a floor, if a person does not enter the elevator and the floor button on the other floor has not been pressed, the elevator closes its door and remains on that floor until another person presses a button on a floor.

For simplicity, assume that all the activities that happen, from when the elevator reaches a floor until the elevator closes its door, take zero time. [*Note*: Although these activities take zero time, they still occur sequentially; e.g., the elevator door must open before the passenger exits the elevator.] The elevator takes five seconds to move from one floor to the other. Once per second, the simulator provides the time to the scheduler and to the elevator. The scheduler and elevator use the time to determine what actions each must take at that particular time, [sic] (*e.g.*, the scheduler might determine that it is time to create a person, and the elevator, if moving, might determine that it is time to arrive at its destination floor).

The simulator should display messages on the screen that describe the

activities that occur in the system. These include a person pressing a floor button, the elevator arriving on a floor, the clock ticking, a person entering the elevator, etc.